needed. For example, transfer of volume information, to permit a disk image to be mounted, only requires the two lowest protocol layers. Consequently, the overhead associated with the higher layers can be avoided, thereby enabling the transfer to take place at extremely high speeds.

This capability is particularly advantageous in the case of graphic files, which typically contain large amounts of data. In one implementation of the invention, an image server can be employed to provide large graphic files to remote sites. Referring to FIG. **7**, the image server has an associated storage medium **80** containing disk images **82** which relate to different respective graphic files. Using low level network protocols, individual images can be mounted at remote client computers **84**. Thus, the client computers can access the information in the image files over the network at extremely high speeds. Alternatively, the image files can be copied from the network server to the client computer by means of low-level network protocols, and then mounted at the client computer. With this approach, large image files can be accessed without having to use higher level network protocols, and be encumbered by the overhead associated with them. Another advantage, discussed previously, is the fact that the image server need not use the same file formatting as the client computers. As long as the disk image drivers at the client computers **84** are capable of reading the file format at the server, the disk images can be mounted as local volumes, independent of the file format employed at the server.

In accordance with another aspect of the invention, disk images can be associated with pre-defined actions to be carried out. For example, it might be desirable to make back-up copies of files stored on a series of floppy disks. Referring to FIG. **8**, files to be replicated are stored on a series of four floppy disks **86**. Each disk may have a checksum (CS) computed for it. A disk image **88** is created for each of the four floppy disks. The checksum is included in the imaged information. Thereafter a script file **90** is created, which lists disk images to be automatically mounted and an action to be taken after the images have been mounted. For the example described above, the script file **90** could identify a disk copy utility application, followed by an identification of each image which is to be mounted and copied. The identification of the application and image files can specify the address at which they are to be found. This address could be a local address or a network address.

When the user actuates the script file, all of the identified images are mounted at the user's computer. If desired, a checksum operation can be carried out with respect to each mounted image, to verify its integrity. The specified application is then launched. In this case, the application creates a backup of each imaged disk on a floppy disk at the user's computer. Once the operation has been completed, the image files can be unmounted, if desired, or remain mounted on the user's computer, depending upon the user's preferences.

In this regard, the disk image driver can mark a mounted image as being "owned" by an application or process, such as the disk copy utility. If an imaged volume is owned, the driver checks at regular intervals whether the owning process is still executing. If the process is no longer executing, the volume is marked as unowned and the driver then unmounts it. If no application program is specified in the script file, the identified disk images are simply mounted in response to launching of the script file and marked as unowned. In this case the user must manually unmount the volumes.

The foregoing aspect of the invention can be employed to create a "onebutton" installer that is not limited in the number and/or size of files that can be installed. Referring to FIG. **9**, disk images **92** are created for one or more volumes containing the files to be installed. These images are then concatenated into a single file **94**, which constitutes an image file tome. Each image in the tome might have an associated index resource. Since all of the files remain intact and separately defined within the respective images **92**, and the images are unmodified in the image-file tome **94**, the original integrity of the files is unaltered. As such, there is no need to requalify any of the files within the tome **94** to ensure their integrity for subsequent installation.

An installation package **96** comprises three elements, namely a conventional installer program **98**, the image file tome **94**, and a script file **100**. The script file identifies the images to be mounted for the installation process, and launches the installer program once the images have been mounted. The program then operates in the normal manner to install the appropriate files from the mounted images onto a computer system. By means of this approach, prior limitations that were placed on installation processes, regarding the number and size of files that can be handled, are overcome, since each image only constitutes a single image file even though it may contain a large number of individual data files. Furthermore, since the user is only required to perform a single action, namely launch the script file, the installation takes place in a true "one-button" manner.

From the foregoing, therefore, it can be seen that the present invention provides an efficient procedure for the dissemination and replication of files in an electronic format which is not limited by the sizes of the files themselves. Through the creation of disk images and mounting of the images at remote computers, ready access is provided to the files in a speedy manner, through the use of low level network communication protocols. Through the availability of different types of formats, backing storage can be used in a manner which is most efficient, taking into account the needs of the user. Furthermore, the mounting and transfer of files is carried out independently of file systems themselves, thereby allowing files to be shared among users of different types of computer system.

It will be appreciated by those of ordinary skill in the art that the present invention can be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The presently disclosed embodiments are therefore considered in all respects to be illustrative and not restrictive. The scope of the invention is indicated by the appended claims rather than the foregoing description, and all changes that come within the meaning and range of equivalence thereof are intended to be embraced therein.

What is claimed:

1. A system for accessing computer-readable files stored on a source device, by a plurality of target computers comprising:

means for creating a disk image of the source device, wherein said source device is a physical storage volume on which said computer-readable files to be accessed by said plurality of target computers are located, and for storing said disk image on a storage device that is accessible to said plurality of target computers, wherein said disk image is a virtual representation of said physical storage volume such that it includes volume format information that reflects the format of said physical storage volume, and which enables said disk image to be mounted at each of said plurality of target computers; and

a disk image driver at each of said plurality of target computers having access to file format information

which enables said target computers to read files, which exhibit different file formats, contained on said disk image.

2. The system of claim 1 wherein said disk image driver includes an index which identifies correspondence between address location in said storage volume and address locations in said disk image.

3. The system of claim 2 wherein said disk image contains a compressed version of data in said files, and wherein said index further includes information pertaining to the manner in which the data was compressed.

4. The system of claim 3 wherein the data in said disk image is divided into individual chunks which are separately compressed and said index contains, for each chunk, the address of the chunk of data in the file, the address for the corresponding compressed data in the disk image, and an identification of a compression algorithm via which the data of that chunk was compressed.

5. The system of claim 4 wherein different chunks of data are compressed via different respective algorithms.

6. The system of claim 4 wherein different chunks of data have different respective sizes.

7. The system of claim 1 wherein said disk image driver includes data pertaining to different types of file systems, to thereby enable said disk image driver to access disk images stored in different disk image file formats respectively related to said different types of file systems.

8. The system of claim 1, wherein said disk image is stored on the storage device in a compressed read/only format comprising a file which contains compressed versions of chunks of data stored in said physical storage volume, and an index which provides a mapping between logical address blocks in said physical storage volume and addresses of corresponding compressed data in said file.

9. The system of claim 8, wherein said index contains information pertaining to the manner in which the chunks of data were compressed.

10. The system of claim 9, wherein the data in said volume is divided into individual chunks which are separately compressed and said index contains, for each chunk, the address of the chunk of data in said physical storage volume, the address for the corresponding compressed data in said disk image, and an identification of a compression algorithm via which the data of that chunk was compressed.

11. The system of claim 10, wherein different chunks of data are compressed via different respective algorithms.

12. The system of claim 10, wherein different chunks of data have different respective sizes.

13. The system of claim 10, wherein said uncompressed read/only format also has an associated index which provides a mapping between logical address blocks in said physical storage volume and addresses of corresponding data in the file.

14. The system of claim 1, wherein said disk image is stored on the storage device in a read/write format comprising a file which contains a copy of every logical address block in said physical storage volume, regardless of whether the blocks contain data.

15. The system of claim 1, wherein said disk image is stored on the storage device in an uncompressed read/only format comprising a file which contains volume information and a copy of only those logical address blocks of the physical storage volume which contain data.

16. A method for providing a remote computer access to files stored on a source device, comprising the steps of:

creating a disk image of said source device, wherein said source device is a physical storage volume which contains said files to be accessed by said remote computer, and wherein said disk image is a virtual representation of said physical storage volume in that said disk image includes volume format information that reflects the format of said physical storage volume;

generating a script file which includes an identification of said disk image;

launching said script file at said remote computer; and

mounting, at said remote computer, the disk image identified in said script file using a disk image driver that has access to volume format information which is needed to mount files, exhibiting different file formats, on the disk image.

17. The method of claim 16 wherein said script file also includes an identification of an executable program, and further including the step of running said program at the remote computer after mounting said disk image.

18. The method of claim 17 wherein said program is an installer program which installs files from the mounted disk image onto the remote computer.

19. The method of claim 16 wherein a plurality of disk images are created and identified in said script file, and wherein all of the disk images identified in said script file are mounted at said remote computer.

20. The method of claim 16 further comprising the step of:

selectively storing said disk image in a storage medium device in any one of the following disk image file formats:

a read/write format comprising a file which contains a copy of every logical address block in said physic al storage volume, regardless of whether the blocks contain data;

an uncompressed read/only format comprising a file which contains volume information and a copy of only those logical address blocks of said physical storage volume which contain data; and

a compressed read/only format comprising a file which contains compressed versions of chunks of data stored in said physical storage volume, and an index which provides a mapping between logical address blocks in said physical storage volume and addresses of corresponding compressed data in said file.

* * * * *